

1. Download all the data in this folder <https://drive.google.com/open?id=124TV17FcFVEx8Qd14j09qvxvaqlSqoEu>. It contains two file both images and labels. The label file list the images and their categories in the following format:

path/to/the/image.tif,category

where the categories are numbered 0 to 15, in the following order:

- 0 letter
- 1 form
- 2 email
- 3 handwritten
- 4 advertisement
- 5 scientific report
- 6 scientific publication
- 7 specification
- 8 file folder
- 9 news article
- 10 budget
- 11 invoice
- 12 presentation
- 13 questionnaire
- 14 resume
- 15 memo

2. On this image data, you have to train 3 types of models as given below. You have to split the data into Train and Validation data.

3. Try not to load all the images into memory, use the gennarators that we have given the reference notebooks to load the batch of images only during the train data. or you can use this method also <https://medium.com/@vijayabhaskar96/tutorial-on-keras-imagedatagenerator-with-flow-from-dataframe-8bd5776e45c1>

<https://medium.com/@vijayabhaskar96/tutorial-on-keras-flow-from-dataframe-1fd493d237c>

4. You are free to choose Learning rate, optimizer, loss function, image augmentation, any hyperparameters. but you have to use the same architecture what we are asking below.

5. Use tensorboard for every model and analyse your gradients. (you need to upload the screenshots for each model for evaluation)

Note: fit_generator() method will have problems with the tensorboard histograms, try to debug it, if you could not do use histgrams=0 i.e don't include histograms, check the documentation of tensorboard for more information.

6. You can check about Transfer Learning in this link - <https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html>

Model-1

1. Use VGG-16 pretrained network without Fully Connected layers and initialize all the weights with Imagenet trained weights.
2. After VGG-16 network without FC layers, add a new Conv block (1 Conv layer and 1 Maxpooling), 2 FC layers and a output layer to classify 16 classes. You are free to choose any hyperparameters(parameters of conv block, FC layers, output layer.
3. Final architecture will be **INPUT --> VGG-16 without Top layers(FC) --> Conv Layer --> Maxpool Layer --> 2 FC layers --> Output Layer**
4. Train only new Conv block, FC layers, output layer. Don't train the VGG-16 network.

```
In [31]: # %tensorflow_version 2.x

UsageError: Line magic function '%tensorflow_version' not found.

In [33]: # !pip uninstall keras-preprocessing

In [41]: # !pip install git+https://github.com/keras-team/keras-preprocessing.git

In [31]: import tensorflow as tf
import os
import numpy as np
import pandas as pd
import keras
import cv2
import datetime
import random as rn

import tensorflow.keras.models import *
from tensorflow.keras.layers import *
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Sequential,Model
from tensorflow.keras.layers import Dense,MaxPooling2D,MaxPooling2D
from tensorflow.keras.layers import Flatten,Dropout,concatenate,Input
from tensorflow.keras.layers import Embedding
from tensorflow.keras.layers import Conv2D,Conv2D
from tensorflow.keras.callbacks import TensorBoard,Callback
from tensorflow.keras.callbacks import EarlyStopping
%reload_ext tensorboard

from tensorflow.keras.utils import plot_model
from keras_preprocessing.image import ImageDataGenerator

import pydot_ng as pydot
pydot.find_graphviz()
```

```
In [41]: images = pd.read_csv('labels.final.csv')
images['label'] = images['label'].astype(str)
images_directory = 'data.final'
```

```
In [51]: target_size = (128, 128)
```

```
In [61]: datagen = ImageDataGenerator(rescale=1./255,validation_split=0.2)
```

```
In [71]: train_generator = datagen.flow_from_dataframe(images, images_directory, x_col='path',y_col='label',class_mode='categorical', target_size=target_size, batch_size=32)
Found 88400 validated image filenames belonging to 16 classes.
```

```
In [81]: validation_generator = datagen.flow_from_dataframe(images, images_directory, x_col='path',y_col='label',class_mode='categorical', target_size=target_size, batch_size=32)
Found 9600 validated image filenames belonging to 16 classes.
```

```
In [331]: tf.keras.backend.clear_session()
np.random.seed(0)
rn.seed(0)

base_model = tf.keras.applications.vgg16.VGG16(weights='imagenet', include_top=False, input_shape=(target_size, 3))

for layer in base_model.layers:
    layer.trainable = False

conv2d_1 = Conv2D(filters = 512, kernel_size=(3, 3), activation='relu')(base_model.output)
pool_1 = MaxPooling2D()(conv2d_1)

flatten = Flatten(data_format='channels_last',name='Flatten')(pool_1)
dense_1 = Dense(units = 2048, activation='relu')(flatten)
dense_2 = Dense(units = 512, activation='relu')(dense_1)
out = Dense(units = 16, activation='softmax')(dense_2)
model_1 = Model(inputs=base_model.inputs, outputs=out)
model_1.compile(optimizer='Adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

```
In [341]: log_dir = logs/fit+"model_1." + datetime.datetime.now().strftime("%Y%m%d-%H%M%S")
tensorboard_callback = tf.keras.callbacks.TensorBoard(log_dir=log_dir,histogram_freq=1, write_graph=True,write_grads=True)
early_stopping = keras.callbacks.EarlyStopping(patience=2)

WARNING:tensorflow: 'write_grads' will be ignored in TensorFlow 2.0 for the 'TensorBoard' Callback.
```

```
In [51]: class AchieveAccuracy(Callback):
def __init__(self, target):
    super(AchieveAccuracy, self).__init__()
    self.target = target
def on_epoch_end(self, epoch, logs={}):
    acc = logs['accuracy']
    if acc == self.target:
        self.model.stop_training = True

In [61]: %time
batch_size = 256
steps = len(train_generator.labels) // batch_size
target = AchieveAccuracy(0.6)
callbacks = [early_stopping,tensorboard_callback,target]
print(model_1.summary())
```

```
Model: "model1"

Layer (type) Output Shape Param #
=====
input_1 (InputLayer) [(None, 128, 128, 3)] 0
block1_conv1 (Conv2D) (None, 128, 128, 64) 1792
block1_conv2 (Conv2D) (None, 128, 128, 64) 36928
block1_pool (MaxPooling2D) (None, 64, 64, 64) 0
block2_conv1 (Conv2D) (None, 64, 64, 128) 73856
block2_conv2 (Conv2D) (None, 64, 64, 128) 147584
block2_pool (MaxPooling2D) (None, 32, 32, 128) 0
block3_conv1 (Conv2D) (None, 32, 32, 256) 295168
block3_conv2 (Conv2D) (None, 32, 32, 256) 590880
block3_conv3 (Conv2D) (None, 32, 32, 256) 590880
block3_pool (MaxPooling2D) (None, 16, 16, 256) 0
block4_conv1 (Conv2D) (None, 16, 16, 512) 1180160
block4_conv2 (Conv2D) (None, 16, 16, 512) 2359808
block4_conv3 (Conv2D) (None, 16, 16, 512) 2359808
block4_pool (MaxPooling2D) (None, 8, 8, 512) 0
block5_conv1 (Conv2D) (None, 8, 8, 512) 2359808
block5_conv2 (Conv2D) (None, 8, 8, 512) 2359808
block5_conv3 (Conv2D) (None, 8, 8, 512) 2359808
block5_pool (MaxPooling2D) (None, 4, 4, 512) 0
conv2d (Conv2D) (None, 2, 2, 512) 2359808
max_pooling2d (MaxPooling2D) (None, 1, 1, 512) 0
Flatten (Flatten) (None, 512) 0
dense (Dense) (None, 2048) 1050624
dense_1 (Dense) (None, 512) 1049888
dense_2 (Dense) (None, 16) 8208
=====
Total params: 19,182,416
Trainable params: 4,407,728
Non-trainable params: 14,714,688

None
Wall time: 6.95 ms
```

```
In [261]: history_1 = model_1.fit(train_generator, validation_data = validation_generator,epochs=50, steps_per_epoch=steps,callbacks=callbacks)

Epoch 1/50
150/150 [=====] - 1410s 9s/step - loss: 1.3127 - accuracy: 0.5973 - val_loss: 1.3151 - val_accuracy: 0.5969
Epoch 2/50
150/150 [=====] - 1311s 9s/step - loss: 1.2653 - accuracy: 0.6058 - val_loss: 1.2765 - val_accuracy: 0.6056
```

```
In [271]: print("Accuracy->",(history_1.history['accuracy'])[-1])*100)

Accuracy-> 60.583335161209106
```

```
In [281]: %reload_ext tensorboard
%tensorboard --logdir=logs/fit/'

Reusing TensorBoard on port 6006 (pid 444), started 1:00:53 ago. (Use 'kill 444' to kill it.)
```

ERROR

The requested URL could not be retrieved

The following error was encountered while trying to retrieve the URL: <http://127.0.0.1:6006/>

Access Denied.

Access control configuration prevents your request from being allowed at this time. Please contact your service provider if you feel this is incorrect.

Your cache administrator is [webmaster](#).

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)

Generated Sat, 04 Sep 2021 06:57:57 GMT by httpd/2.4.18 (Ubuntu)